

NAME

inchstr, **inchnstr**, **winchstr**, **winchnstr**, **mvinchstr**, **mvinchnstr**, **mvwinchstr**, **mvwinchnstr** – get a *curses* character string from a window

SYNOPSIS

```
#include <curses.h>

int inchstr(chtype * chstr);
int inchnstr(chtype * chstr, int n);
int winchstr(WINDOW * win, chtype * chstr);
int winchnstr(WINDOW * win, chtype * chstr, int n);

int mvinchstr(int y, int x, chtype * chstr);
int mvinchnstr(int y, int x, chtype * chstr, int n);
int mvwinchstr(WINDOW * win, int y, int x, chtype * chstr);
int mvwinchnstr(WINDOW * win, int y, int x, chtype * chstr, int n);
```

DESCRIPTION

winchstr extracts a *curses* character string from a *curses* window *win*, starting at the cursor and stopping at the end of the line, and stores it in *chstr*, terminating it with a null *curses* character. **winchnstr** does the same, but copies at most *n* *curses* characters from *win*. A negative *n* implies no limit; **winchnstr** then works like **winchstr**. **ncurses(3X)** describes the variants of these functions.

RETURN VALUE

These functions return **OK** on success and **ERR** on failure.

In *ncurses*, these functions fail if

- the *curses* screen has not been initialized,
- (for functions taking a *WINDOW* pointer argument) *win* is a null pointer, or
- *chstr* is a null pointer.

Functions prefixed with “mv” first perform cursor movement and fail if the position (*y*, *x*) is outside the window boundaries.

NOTES

All of these functions except **winchnstr** may be implemented as macros.

Reading a line that overflows the array pointed to by *chstr* and its variants causes undefined results. Instead, use the *n*-infix functions with a positive *n* argument no larger than the size of the buffer backing *chstr*.

EXTENSIONS

inchnstr, **winchnstr**, **mvinchnstr**, and **mvwinchnstr**’s acceptance of negative *n* values is an *ncurses* extension.

PORTABILITY

Applications employing *ncurses* extensions should condition their use on the visibility of the **NCURSES_VERSION** preprocessor macro.

X/Open Curses Issue 4 describes these functions. It specifies no error conditions for them. It characterizes the strings stored by these functions as containing “at most *n* elements” from a window, but does not specify whether the string stored by these functions is null-terminated.

SVr4 does not document whether it null-terminates the *curses* character string it stores in *chstr*, and does not document whether a trailing null *curses* character counts toward the length limit *n*.

SVr4 describes a successful return value only as “an integer value other than *ERR*”.

HISTORY

SVr3.1 (1987) introduced these functions.

SEE ALSO

curs_in_wchstr(3X) describes comparable functions of the *ncurses* library in its wide-character configuration (*ncursesw*).

curses(3X), **curs_inch**(3X), **curs_inwstr**(3X)